FADAS: Towards Federated Adaptive Asynchronous Optimization

Yujia Wang[1], Shiqiang Wang[2], Songtao Lu[2], Jinghui Chen[1]

[1]The Pennsylvania State University [2]IBM Research

Full Paper:

Federated Learning (FL)

General FL ERM objective:

$$\min_{x \in R^d} f(x) := \frac{1}{N} \sum_{i=1}^{N} F_i(x) = \frac{1}{N} \sum_{i=1}^{N} E_{\xi_i \sim D_i}[F_i(x; \xi_i)]$$

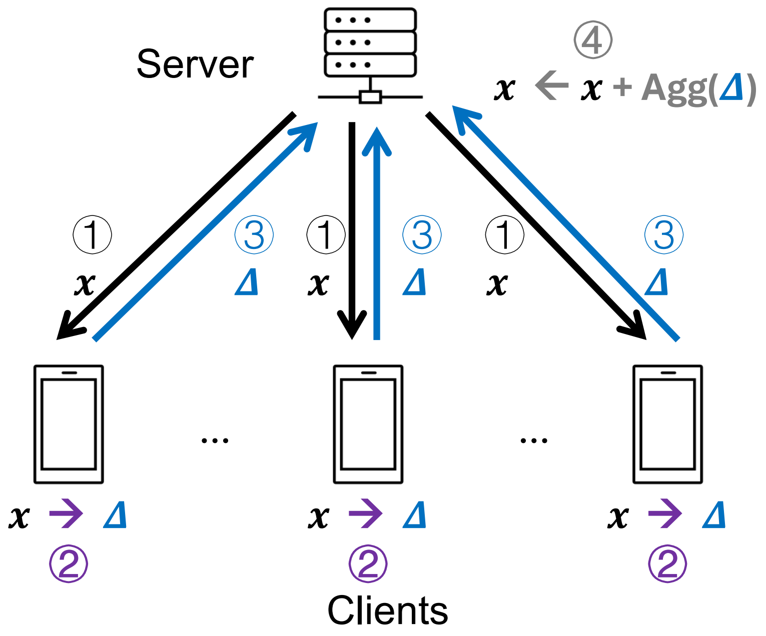
Steps of FL:

① Server: broadcasts global model x to selected clients

② Clients: local training for K steps
and get model difference Δ

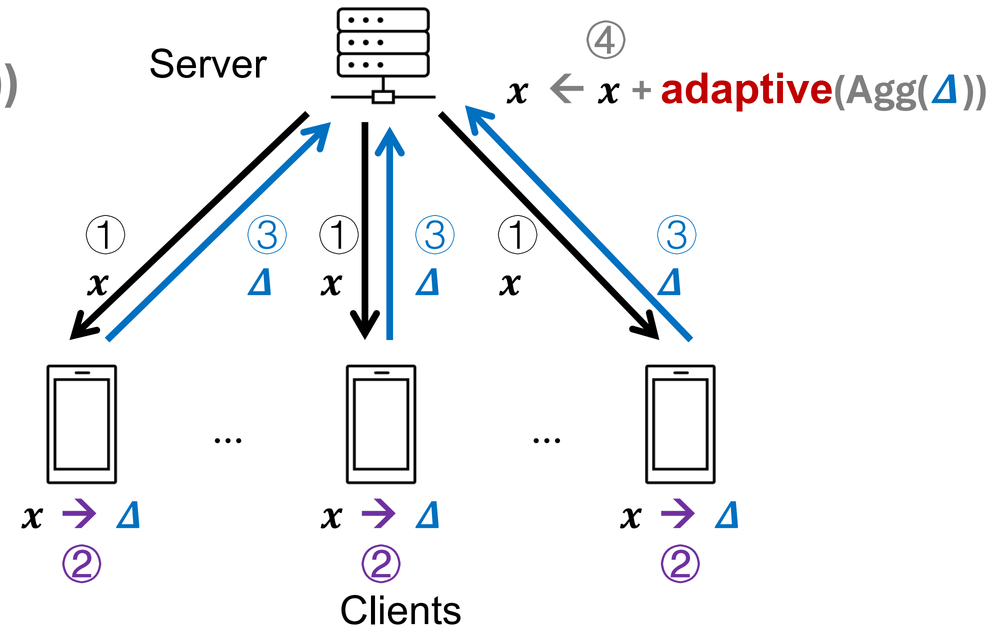
③ Clients: upload model difference Δ to the server

④ Global model aggregation and update (FedAvg,
FedProx, FedAMS, etc.)



Adaptive Federated Optimization

- Adaptive optimization shows the advantage over SGD in many cases, e.g., training/fine-tuning large-scale models
- Incorporating adaptive optimization into FL:
  - Server: take the Agg(Δ) as a **pseudo-gradient**
  - Apply adaptive optimizer: $x \leftarrow x + \textbf{adaptive}(\text{Agg}(\Delta))$



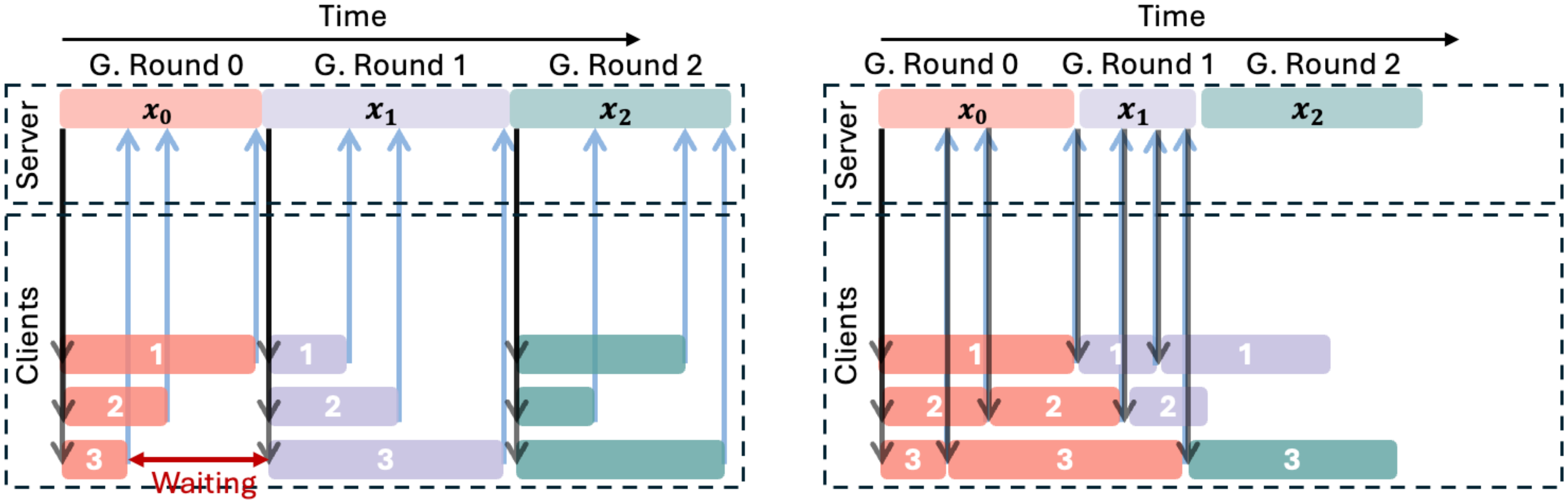
Adaptive Federated Optimization

- However, existing adaptive FL methods rely on traditional synchronous aggregation:
  - **Clients update at different speeds** due to variable computation and communication capabilities
  - **Server needs to wait** for all participating clients to complete their local training before global updates

Asynchronous Updates for Adaptive Federated Optimization

- Asynchronous updates improve the training efficiency:

  Clients update at their own pace; not required to wait for slower ones



Synchronous Asynchronous

↓: Global model broadcasted from the server to clients

↑: Clients update to the server

FADAS: Federated Adaptive Asynchronous Optimization

How to develop **an asynchronous method for adaptive federated optimization** (with provable guarantees) that enhances training efficiency and is resilient to asynchronous delays?

Global adaptive optimization

+

Delay-adaptive learning rate

FADAS: Federated Adaptive Asynchronous Optimization

- Adopts an asynchronous training scheme, with the concept of **concurrency** (the number of clients that are actively performing local training) and **buffer size** (the number of accumulated updates)

- **Global adaptive optimization**

After the server aggregates to obtain model update difference Δ_t, it updates via

$$\begin{cases} \boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{\Delta}_t, \\ \boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{\Delta}_t \odot \boldsymbol{\Delta}_t, \\ \widehat{\boldsymbol{v}}_t = \max(\widehat{\boldsymbol{v}}_{t-1}, \boldsymbol{v}_t). \end{cases} \tag{3}$$

FADAS: Federated Adaptive Asynchronous Optimization

- **Delay tracking**

  The **server tracks the delay**: $\boldsymbol{x}_{t'}$ is sent to client i at communication round t', and Δ_t^{i} is received at communication round t

  → the gradient delay for Δ_t^{i} is $\tau_t^i = t - t'$

- **Delay-adaptive learning rate**

  The received model updates at communication round t have a maximum delay of

  - $\tau_t^{\max} := \max\{\tau_t^{\mathrm{i}}, i \in M_t\}$,

  where clients in M_t update to the server.

  With a delay threshold τ_c, define a **delay-adaptive learning rate** as in Eq. (4)

❖ **Turn the learning rates down** for the model update $\boldsymbol{\Delta}_t^{\mathrm{i}}$ with larger delays.

❖ If $\tau_t^{\max} > \tau_c$, scale η_t down to **avoid updates with high latency worsening convergence**

$$\eta_t = \begin{cases} \eta & \text{if } \tau_t^{\max} \le \tau_c, \\ \min\left\{\eta, \frac{1}{\tau_t^{\max}}\right\} & \text{if } \tau_t^{\max} > \tau_c. \end{cases} \tag{4}$$

Convergence Analysis

- **Standard FADAS without delay adaptation** (assumptions of smoothness, bounded variance, bounded gradient, bounded delay, and uniform arrivals are assumed):

Corollary A.2. *If we choose the global learning rate $\eta = \Theta(\sqrt{M})$ and $\eta_l = \Theta\left(\frac{\sqrt{\mathcal{F}}}{\sqrt{TK(\sigma^2+K\sigma_g^2)}}\right)$ in Theorem A.1, then for sufficiently large T, the global iterates $\{\boldsymbol{x}_t\}_{t=1}^{T}$ of Algorithm 1 satisfy*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\boldsymbol{x}_t)\|^2] \le \mathcal{O}\left(\boxed{\frac{\sqrt{\mathcal{F}}\sigma}{\sqrt{TKM}} + \frac{\sqrt{\mathcal{F}}\sigma_g}{\sqrt{TM}} + \frac{\mathcal{F}}{T}} + \boxed{\frac{\mathcal{F}G}{T\sqrt{M}}} + \boxed{\frac{\mathcal{F}\tau_{\max}\tau_{\text{avg}}}{T}}\right),$$

where $\mathcal{F} = f(\boldsymbol{x}_1) - f_*$, $f_* = \min_{\boldsymbol{x}} f(\boldsymbol{x}) > -\infty$.

Standard in FL rates

Standard in adaptive FL rates

$\tau_{\max}$: maximum delay
τ_{avg}: average of the maximum delay over time

❖ Compared with the convergence rate of FedBuff in [a] and [b], FADAS obtains a relaxed dependency on the worst-case gradient delay $\tau_{\max}$

❖ When $\tau_{\max}$ is large, the last term becomes the dominant term in the convergence rate

→ **A large worst-case delay τ_{max} may lead to a worse convergence rate**

[a] Nguyen, John, et al. "Federated learning with buffered asynchronous aggregation." International Conference on Artificial Intelligence and Statistics. PMLR, 2022.
[b] Toghani, Mohammad Taha, and César A. Uribe. "Unbounded gradients in federated learning with buffered asynchronous aggregation." *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2022.

Convergence Analysis

- **Delay-adaptive FADAS**

τ_{median}: the median of the maximum delay over all communication rounds T

Corollary A.3. *If we pick $\tau_c = \tau_{\text{median}}$, the global learning rate $\eta = \Theta(\sqrt{M}/\tau_c)$ and $\eta_l = \Theta\left(\frac{\tau_c\sqrt{\mathcal{F}}}{\sqrt{TK(\sigma^2+K\sigma_g^2)}}\right)$, then for sufficiently large T, the global iterates $\{x_t\}_{t=1}^T$ of Algorithm 1 satisfy*

$$\frac{1}{\sum_{t=1}^T \eta_t}\sum_{t=1}^T \eta_t \mathbb{E}[\|\nabla f(x_t)\|^2] \leq \mathcal{O}\left(\boxed{\frac{\sqrt{\mathcal{F}}\sigma}{\sqrt{TKM}} + \frac{\sqrt{\mathcal{F}}\sigma_g}{\sqrt{TM}}} + \boxed{\frac{\mathcal{F}G\tau_c}{T\sqrt{M}} + \frac{\mathcal{F}\tau_{\text{avg}}}{T} + \frac{\mathcal{F}(\tau_c^2 + \tau_c\tau_{\text{avg}})}{T}}\right),$$

where $\mathcal{F} = f(x_1) - f_*$, $f_* = \min_x f(x) > -\infty$. Standard in FL rates **Delay related but does not rely on $\tau_{\max}$!**

❖ The convergence rate here **does not rely on the (possibly large) worst-case delay τ_{max}**

  ❖ Delay-adaptive FADAS is less sensitive to stragglers who may cause a large worst-case delay

❖ When $\tau_c = \tau_{\text{median}} \approx \tau_{\text{avg}} \ll \tau_{\max}$, delay adaptation relaxes the requirement from $\tau_{\max}$ to τ_{median} for achieving the desired convergence rate

Experiments

- Simulate two scenarios: *large worst-case* delay and *mild* delay
- FADAS and its delay-adaptive variant achieve superior test accuracy compared to FedAsync and FedBuff

CIFAR-10, *large worst-case* delay

Method	Dir(0.1) Acc. & std.	Dir (0.3) Acc. & std.
FedAsync	50.92 ± 5.03	75.3 ± 6.18
FedBuff	38.68 ± 8.16	51.32 ± 4.43
FADAS	72.0 ± 0.94	73.27 ± 1.37
FADAS_{da}	$\mathbf{73.96} \pm 3.54$	$\mathbf{79.68} \pm 2.14$

CIFAR-10, *mild* delay

Method	Dir(0.1) Acc. & std.	Dir (0.3) Acc. & std.
FedAsync	42.48 ± 4.93	71.76 ± 3.85
FedBuff	72.15 ± 2.71	79.82 ± 3.25
FADAS	77.68 ± 2.32	82.93 ± 0.81
FADAS_{da}	$\mathbf{78.93} \pm 0.83$	$\mathbf{83.91} \pm 0.54$

GLUE benchmark (selected), *mild* delay

Method	RTE Acc. & std.	MRPC Acc. & std.	SST-2 Acc. & std.
FedAsync	49.46 ± 2.66	69.71 ± 1.02	90.02 ± 0.79
FedBuff	61.61 ± 4.90	76.80 ± 6.05	78.37 ± 4.86
FADAS	64.26 ± 2.30	$\mathbf{83.33} \pm 1.20$	$\mathbf{90.76} \pm 0.26$
FADAS_{da}	$\mathbf{65.10} \pm 2.40$	83.09 ± 1.71	90.05 ± 1.80

Experiments

- Running time comparisons

Training/fine-tuning time simulation, *mild* delay

	Acc.	FedAvg	FedAMS	FADAS	FADAS_{da}
CIFAR-10	75%	2257.7	648.7	**228.0**	237.5
CIFAR-100	50%	1806.3	546.9	**209.8**	**209.8**
RTE	63%	921.9	412.4	**376.2**	436.9
MRPC	80%	1018.1	424.0	**368.3**	370.1
SST-2	90%	-	495.2	73.8	**57.2**

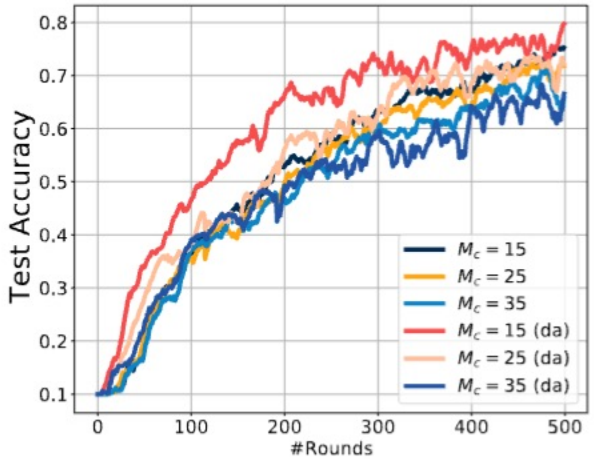
Observation:

❖ In the *large worst-case* delay setting, we observe that $\tau_{\text{avg}} = 10.89$, $\tau_{\text{median}} = 6.0$, and $\tau_{\max} = 127$,
which satisfies $\tau_{\text{median}} \approx \tau_{\text{avg}} \ll \tau_{\max}$ in the previous analysis

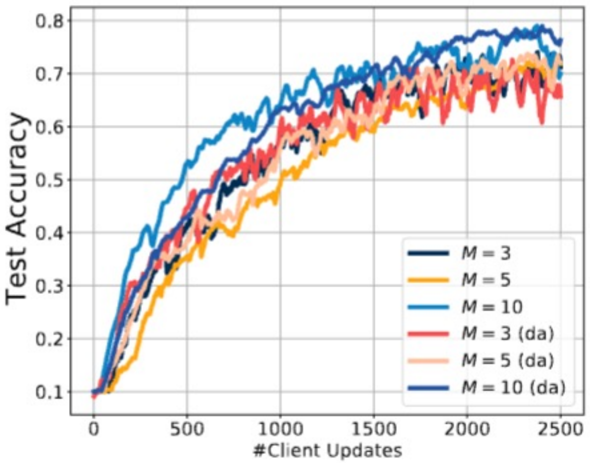
❖ In practice, different thresholds $\tau_c \in \{1,4,8,10\}$ result in similar test accuracy.

Experiments

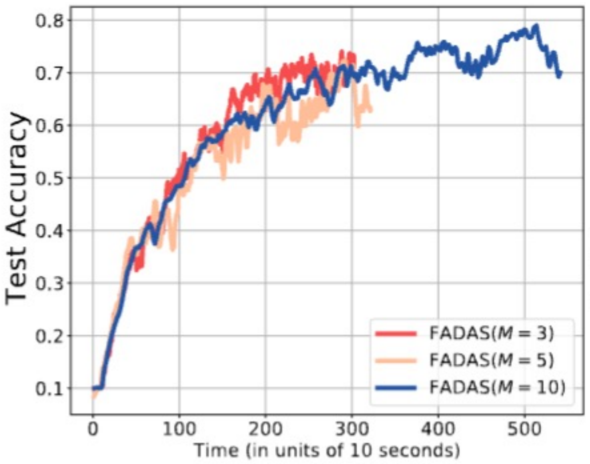
- **Ablation studies** indicate that
  - ❖ smaller concurrency yields better results
  - ❖ larger buffer sizes achieve higher accuracy
  - ❖ smaller buffer sizes require less training time to reach a target accuracy of 70%, particularly in the early stages of training



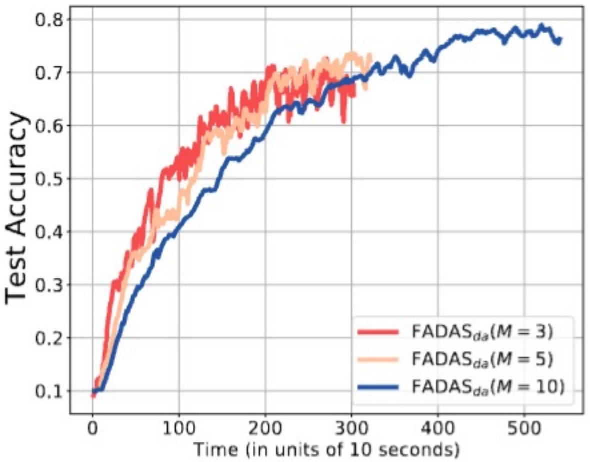
Ablation on concurrency



Ablation on buffer size



Runtime for FADAS



Runtime for FADAS (delay-adaptive)

Thank you!

Please check our full paper through

Welcome to our poster session: **Hall C 4-9 #1208**